Designing a Predictive Coding System for Electronic Discovery

Dhivya Soundarajan, M.S. (HCI) and
Professor Sara Anne Hook, M.B.A., J.D.

HCI International 2017
July 14, 2017

WHAT IS ELECTRONIC DISCOVERY (E-DISCOVERY)?

- Electronic discovery (e-discovery) is something that impacts everyone, whether they know if or not, because it deals with the proper collection, preservation, analysis and production of evidence in digital form.
- To put it bluntly, if you are sued in the U.S., the opposing party's lawyer will be requesting nearly every piece of digital evidence in any format that might be relevant to the case (including social media).
- This presentation will concentrate on the use of predictive coding in civil cases, but e-discovery is part of criminal cases as well as other types of audits and investigations.
- E-discovery is an especially important issue for anyone in Informatics, Media Arts and IT.
- Anyone can find himself/herself needing to comply with requests for potentially relevant evidence – in digital or paper form.

HISTORY OF ELECTONIC DISCOVERY IN THE U.S.

- Series of decisions in *Zubulake v. UBS Warburg* and the 2006 amendments to the Federal Rules of Civil Procedure, a new area within law practice appeared, the law regarding electronic discovery (e-discovery).
- The phase of litigation known as discovery has existed for many years, with opposing parties and their lawyers making requests and exchanging documents that are relevant to a case.
- E-discovery transformed this process from the paper-based, pre-Internet world of discovery to a whole series of rules and decisions related to how to identify, collect, preserve, analyze, review, produce and present electronically-stored information (ESI).
- Efforts to determine standards and best practices, with EDRM being one example, along with the proclamations and guidelines issues by The Sedona Conference.

ELECTRONIC DISCOVERY REFERENCE MODEL (EDRM)

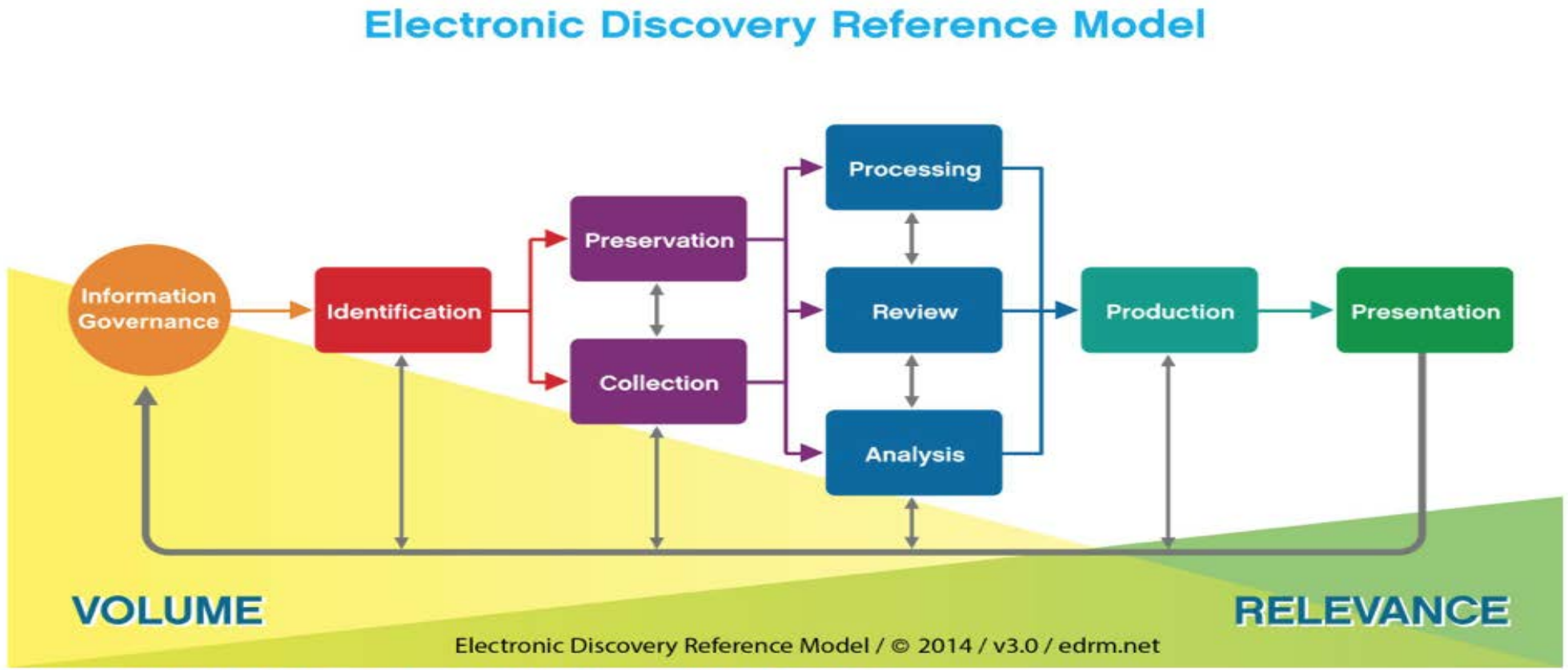


Figure 1 Diagram of the Electronic Discovery Reference Model

See http://www.edrm.net/resources/edrm-stages-explained, accessed 6/29/17.

E-DISCOVERY CHALLENGES

- Not only is this evidence now primarily in digital form, but it also exists a wide range of media and formats, from word processing and spreadsheet files to photographs, blog postings, videos, emails and websites.
- A recent survey conducted by Exterro, Inc., indicated that data volume is still the largest obstacle in e-discovery, with the second biggest obstacle being identifying and accessing sources of ESI.
- E-discovery requests can include social media, text messages and more informal and transient communications, including new services for mobile devices and messaging apps, as well as data from wearable technology (fitness trackers) and the Internet of Things.
- The Federal Rules of Civil Procedure (FRCP), which govern courts in the U.S. federal court system, were revised again in December 2015, with an emphasis on proportionality, streamlining the process and clarification of when and what types of sanctions can be imposed for spoliation of evidence.

WHAT IS PREDICTIVE CODING?

- Predictive coding is the use of keyword search, filtering and sampling to automate portions of an e-discovery process, especially the review stage.
- The goal of predictive coding is to reduce the number of irrelevant and non-responsive ESI that needs to be reviewed manually.
- May also be called – or part of – Technology-Assisted Review (TAR)
- A faulty and incomplete e-discovery process, particularly during the review stage, can result in sanctions and waive the attorney-client privilege or other confidentiality doctrine.
- Predictive coding systems can assist with the overall e-discovery process, leaving humans to concentrate on reviewing the remaining set of ESI before it is produced to the opposing party.
- "[r]esearch shows that human review is far from perfect." *Dynamo Holdings Ltd. P'ship v. Comm'r of Internal Revenue*, WL 4204067 (T.C. July 13, 2016).

COMMON TOOLS IN PREDCTIVE CODING?TAR

- Concept searching
- Contextual searching
- Metadata searching (ESI must usually be produced in native format with the metadata intact)
- Relevance probability and ranking
- Clustering
- Sorting ESI by issues

IS PREDICTIVE CODING ACCEPTED AS PART OF LITIGATION?

- Initially, predictive coding/TAR tools were looked at with considerable suspicion, even though information retrieval, indexing, machine learning and data analytics had been used in other disciplines for many years.
- The reticence to use these types of systems has faded, as illustrated by a long line of cases, starting with the strong support of computer-assisted review articulated in *Da Silva Moore v. Publicis Groupe*, described as the first published opinion recognizing TAR as "an acceptable way to search for relevant ESI in appropriate cases."
- Summaries of recent cases about predictive coding/TAR can be found in The Sedona Conference's new publication, *TAR Case Law Primer*.
- Cases indicate that judge's will likely approve a party's request to use predictive coding, absent some compelling objection.

HOW IS PREDICTIVE CODING USED IN LITIGATION?

- Early case assessment
- Reviewing client ESI before production
  - Prioritizing pre-production review
  - Sorting ESI by potential privilege
  - Quality control – comparing human review with predictive coding results
- Reviewing production from the opposing party
- Other stages of litigation, such as preparing for depositions, responding to summary judgment motions and working with expert witnesses

STATUS OF PREDICTIVE CODING

- "Overall, although the practice of predictive coding is still in its infancy, the number of courts addressing the issue is clearly on the rise. Courts seem to be moving towards permitting, but not requiring, this technology. Litigants that take reasonable positions and strive to work through their disputes with their opponents will typically be much better positioned to prevail in a predictive coding dispute." (Wallis M. Hampton, Predictive Coding: It's Here to Stay. *E-Discovery Bulletin*, June/July 2014, https://www.skadden.com/sites/default/files/publications/LIT_JuneJuly14_EDiscoveryBulletin.pdf, accessed 6/29/17.)
- Note that the support for predictive coding has increased in the past three years since this article was published.

INTRODUCING DHIVYA SOUNDARAJAN

- For nearly two years, Dhivya Soundarajan, a master's-level student in Human-Computer Interaction (HCI), worked with Professor Sara Anne Hook to design a simple predictive coding system based on readily-available software and natural language processing.
- In their paper, Ms. Soundarjajan and Professor Hook describe the purpose of the predictive coding project, the process of developing the system, the software used and what has been designed so far.
- Their paper also discusses proposed future work on the project, including usability testing of the system with a focus group of lawyers who are responsible for e-discovery in their law firms and the features and functionality that they would like to see added to the system.

MODULES ESSENTIAL TO DEVELOP A PREDICTIVE SYSTEMS

Multimodal Input

Store different types of unstructured text, digital archives, emails etc.

Concept Search

An automated information retrieval method to search electronically stored unstructured text which are conceptually similar to the information provided in a search query.

Supervised Machine Learning

The system should not only depend on passive analysis of data but should accept the lawyer's periodic input to enhance the system's efficiency.

MODULES ESSENTIAL TO DEVELOP A PREDICTIVE SYSTEMS

Distributed Storage

Data must be divided into ranges and distributed to multiple servers.

Optimized Storage and Retrieval

Parallel processing of huge amount of data.

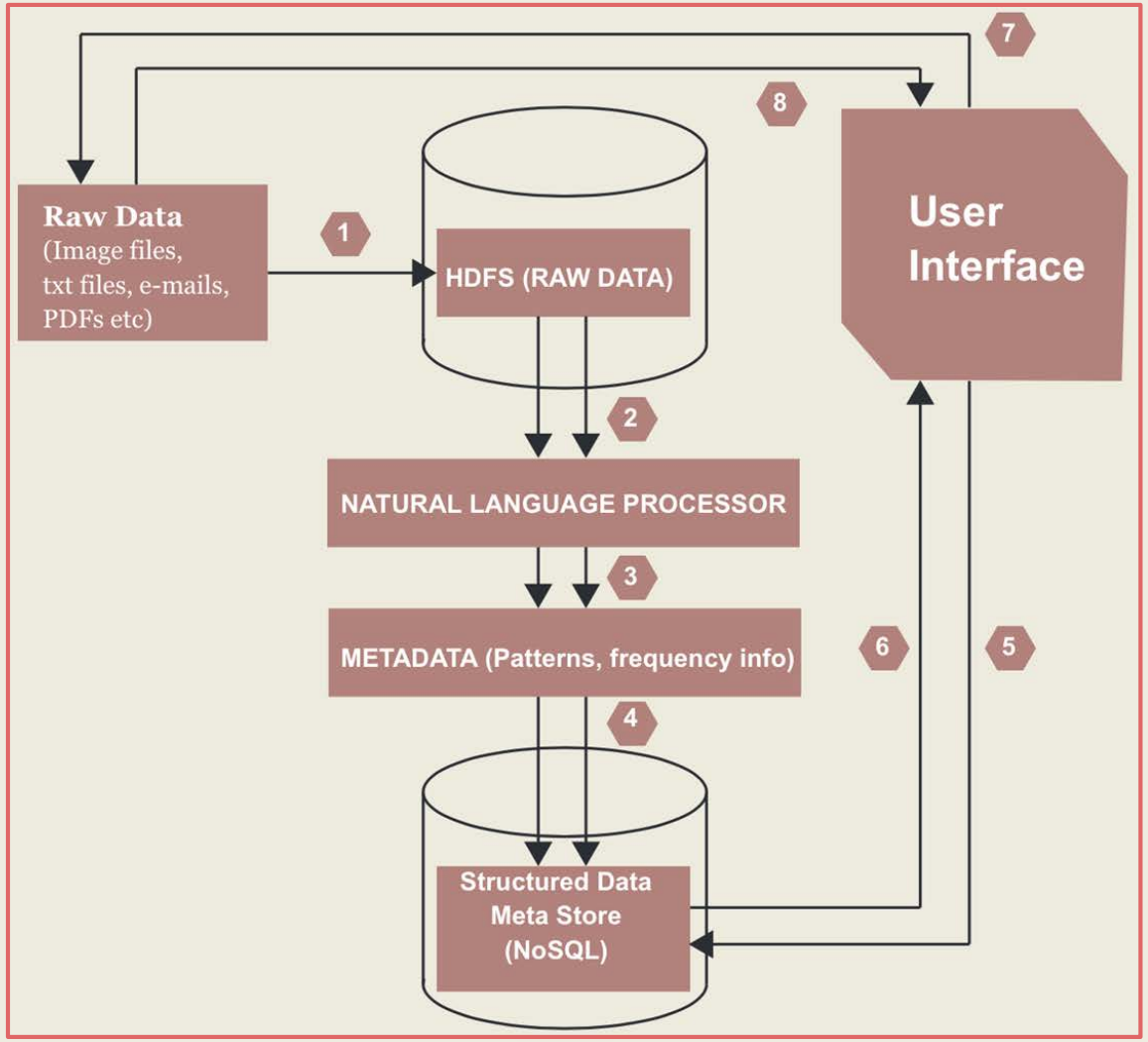
Clean Interface

Interface must be very simple and more usable for novice and expert users. User should able to train the system without any hassle.

SYSTEM ARCHITECTURE

System Architecture:

1. The raw data will be stored in the Hadoop file system along with its location mapping .
2. The data will be extracted for analyzing the patterns.
3. Metadata will be generated for each file based on the computational and processing algorithms.
4. The generated metadata will be stored as structured data in the meta store.
5. The user request will be passed to the meta store
6. The result will be the list containing the documents name and their source location.
7. The user passes the request along with the location parameters
8. The exact files are passed as output to the front-end of the system.



MACHINE LEARNING MODULE - NLP

Train

Use several subsets of files (control sets) that are quintessential, identified by well-trained professionals for both the following cases in order to calibrate the system.

- Positive Sets - relevant files
- Negative Sets - irrelevant files

Then use training sets to train the system.

Analyze

Apply the identified appropriate filters, classifiers and use the pre existing models with tailored specifications to analyze the system.

Evaluate

Check for

- Precision
- Recall
- Measure performance using Extrapolated Precision.*

Since the system is under a supervised learning, system training should happen periodically with new training sets as per the requirement. Then finalize the model for the system.

*Refer to Bill Dimm's blog, https://blog.cluster-text.com/2015/05/19/using-extrapolated-precision-for-performance-measurement/, accessed 6/29/17.

MACHINE LEARNING MODULE - NLP

PRECISION

$$\frac{|(\{relevent\ documents\}) \cap (\{retrieved\ documents\})|}{|\{retrieved\ documents\}|}$$

RECALL

$$\frac{|(\{relevent\ documents\}) \cap (\{retrieved\ documents\})|}{|\{relevent\ documents\}|}$$

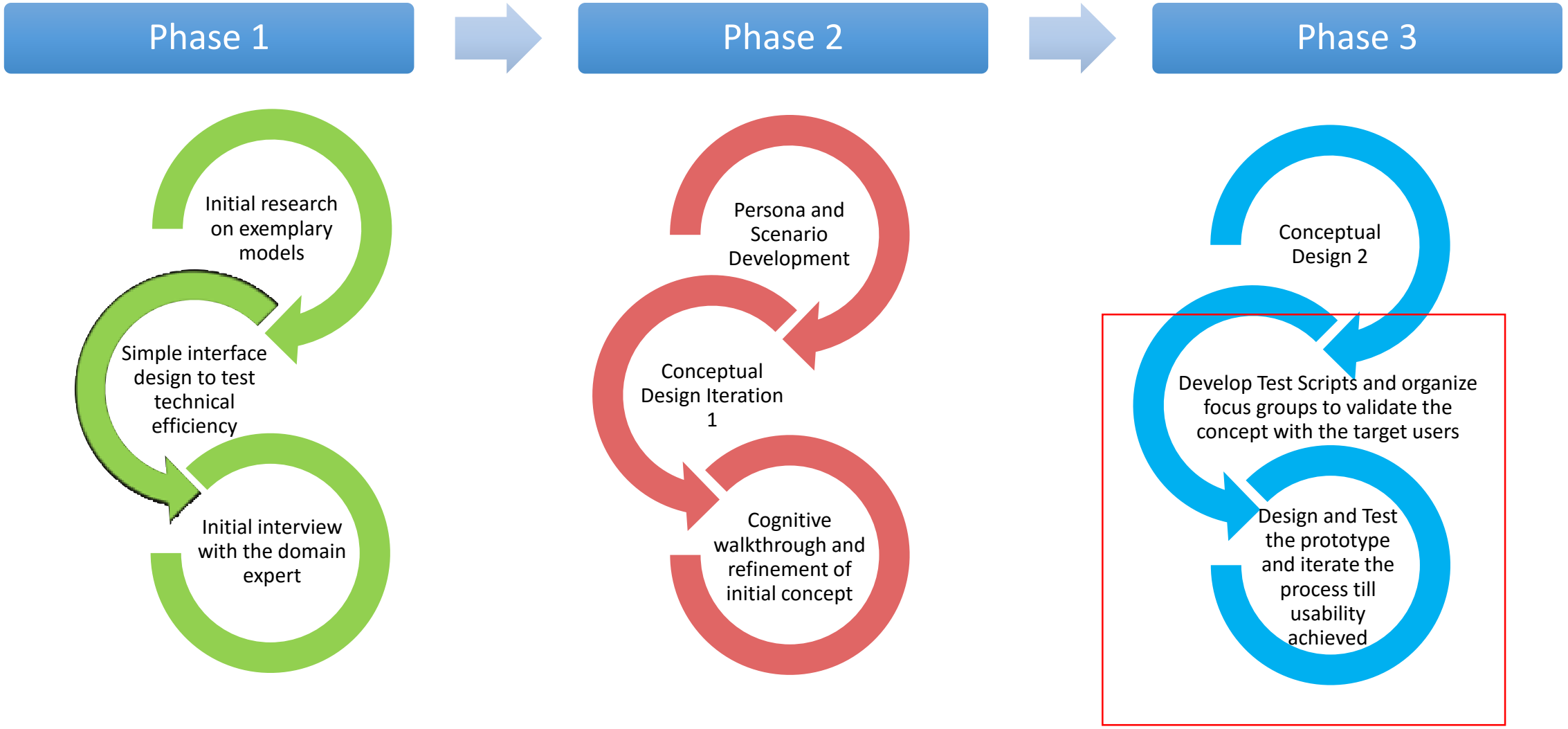
HADOOP DISTRIBUTED FILE SYSTEM & MAP-REDUCE MODULE

Collect different feeds from different nodes (distributed in the cloud). Ex:- Documents, Text messages, Emails etc.

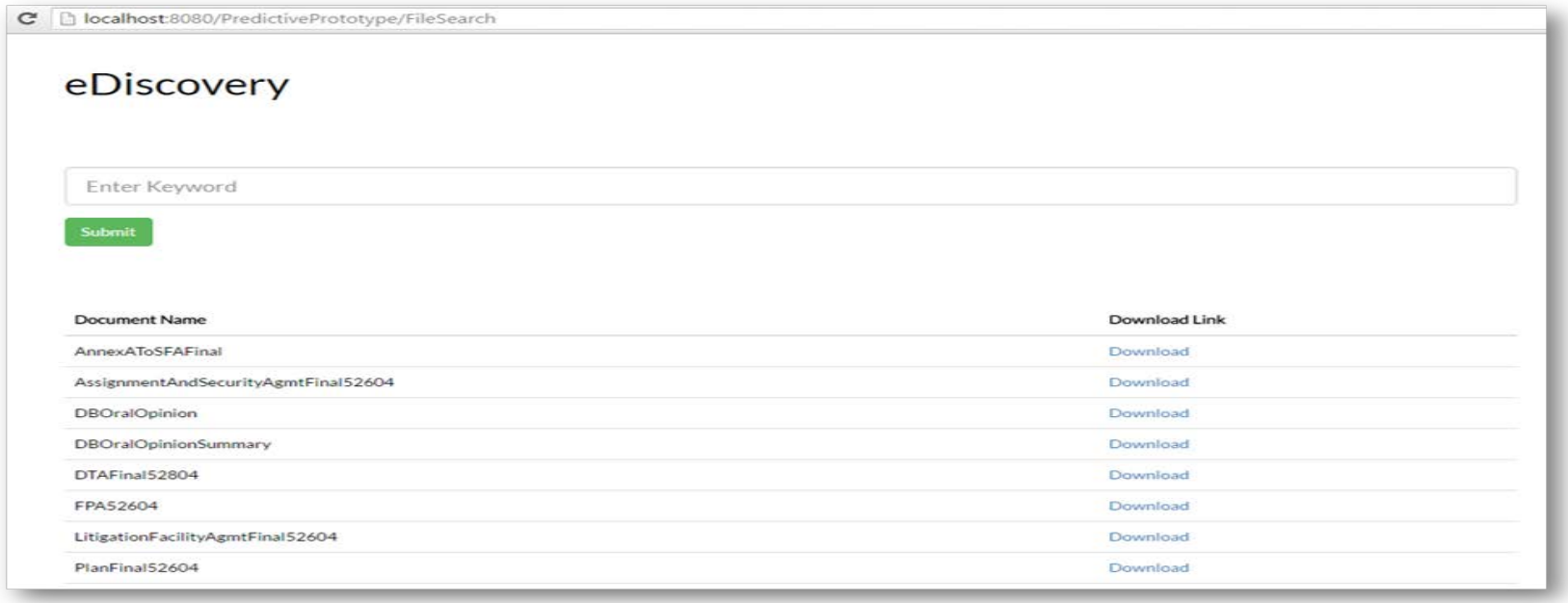
Process data as it flows such as Calculate, Transform, Augment

Display processed files as result of user Query.

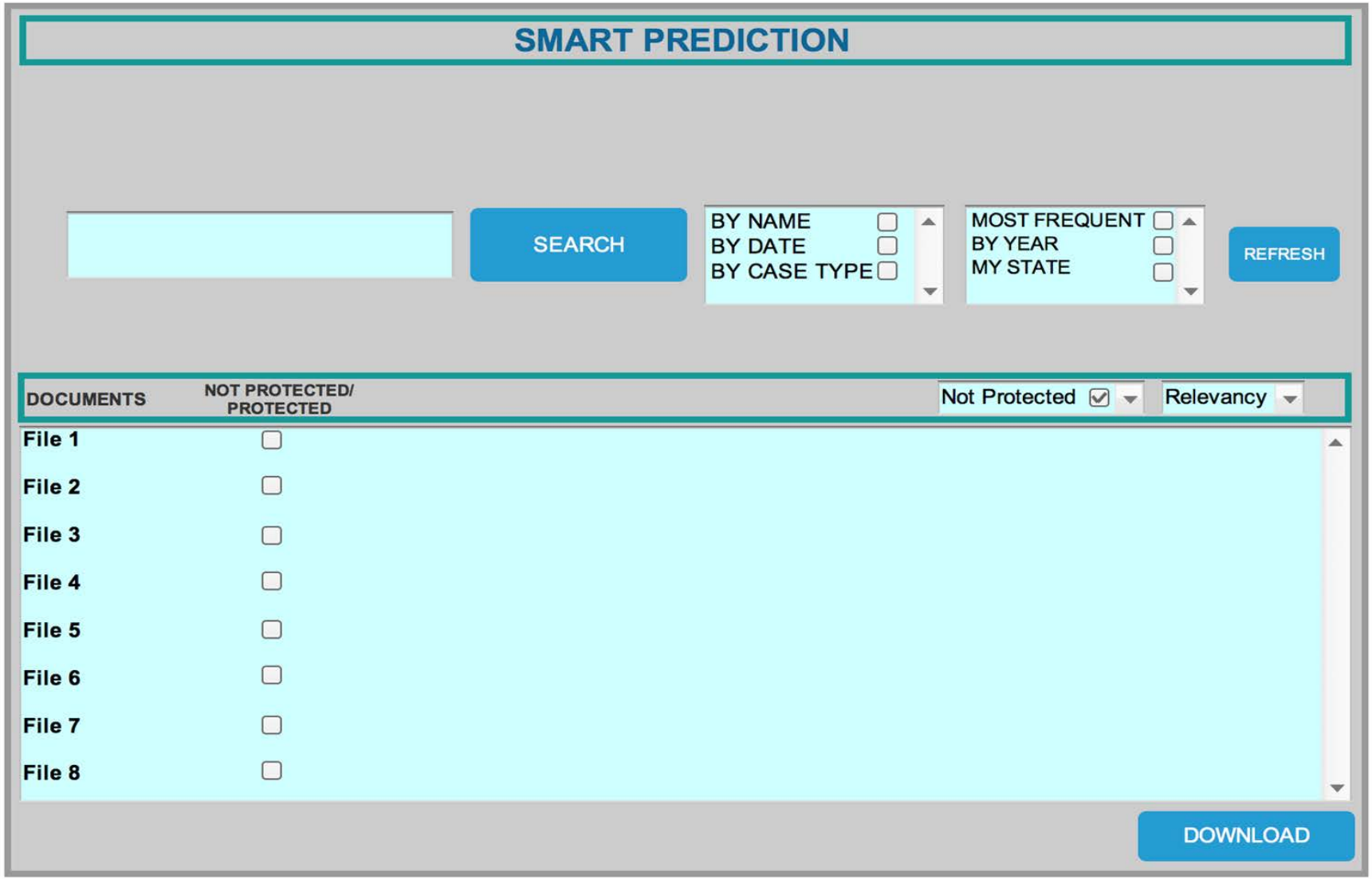
OVERVIEW - UX PROCESS



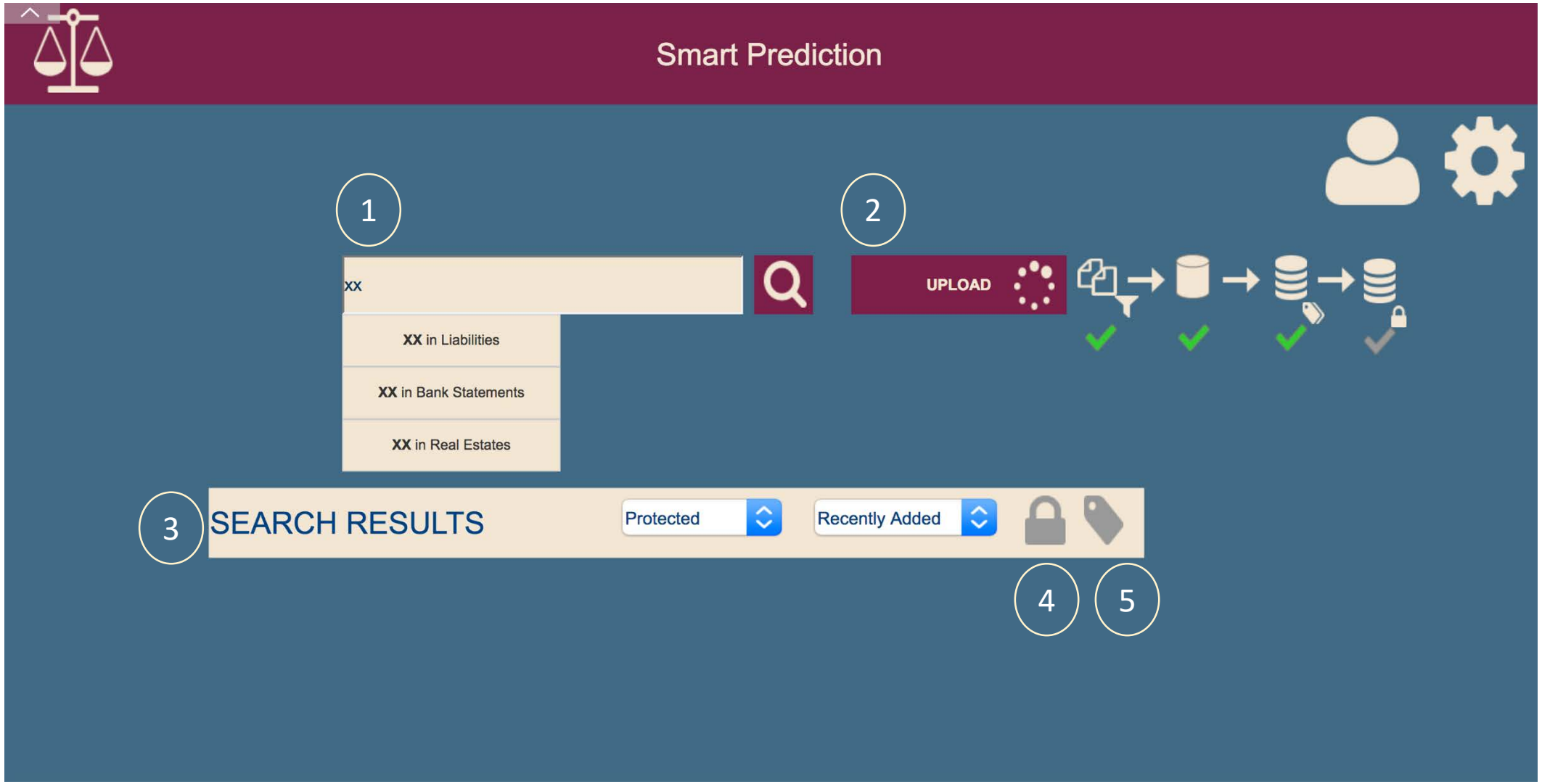
INTERFACE DESIGN MODULE – INITIAL DESIGN



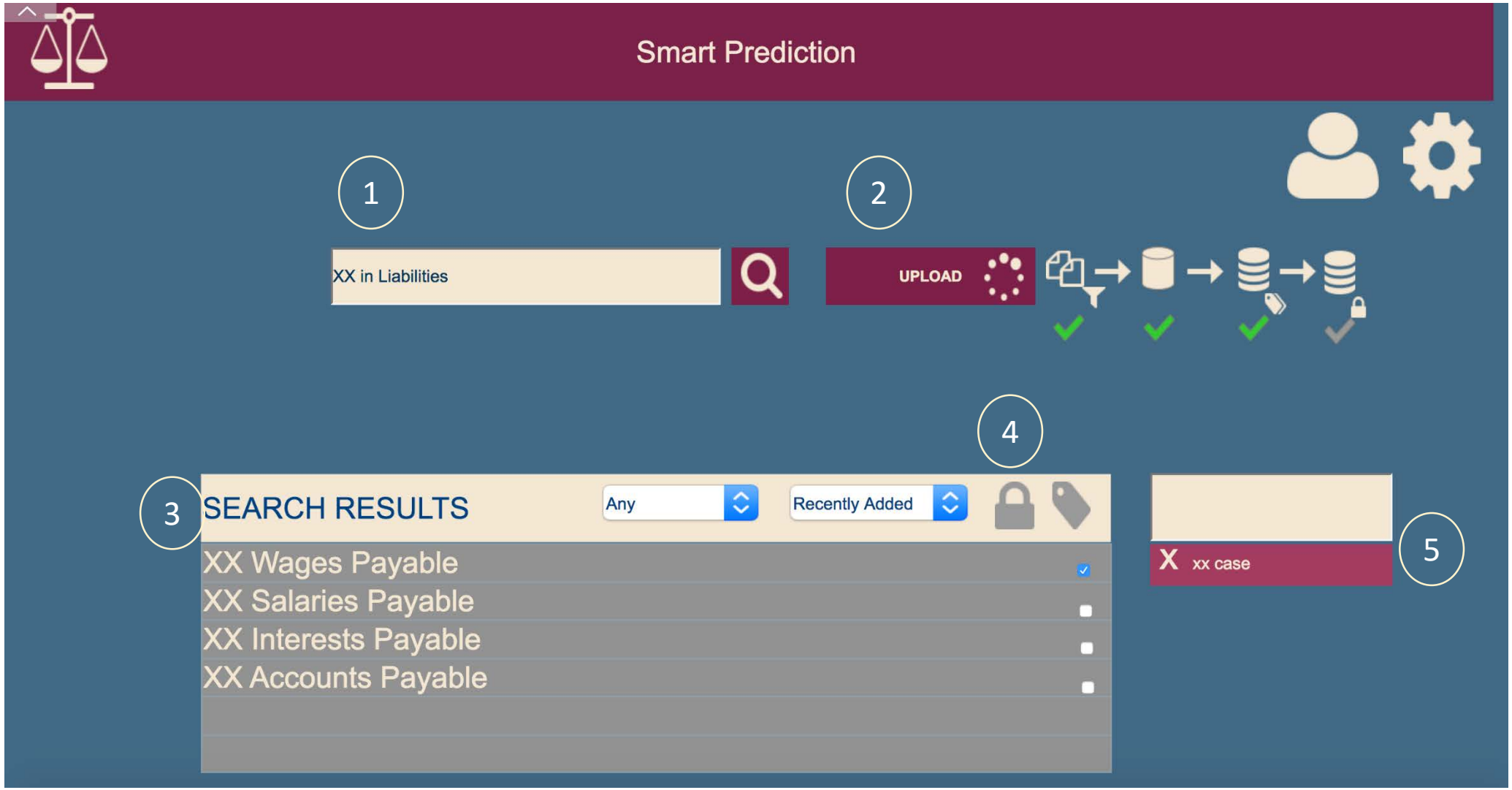
DESIGN ITERATION 2 – CONCEPTUAL DESIGN



DESIGN ITERATION 3



DESIGN ITERATION 3



FOCUS GROUP

DEVELOPED TEXT SCRIPTS

RECRUIT 10 TO 12 PARTICIPANTS

ORGANIZE FOCUS GROUPS IN GROUPS OF 3 TO 4

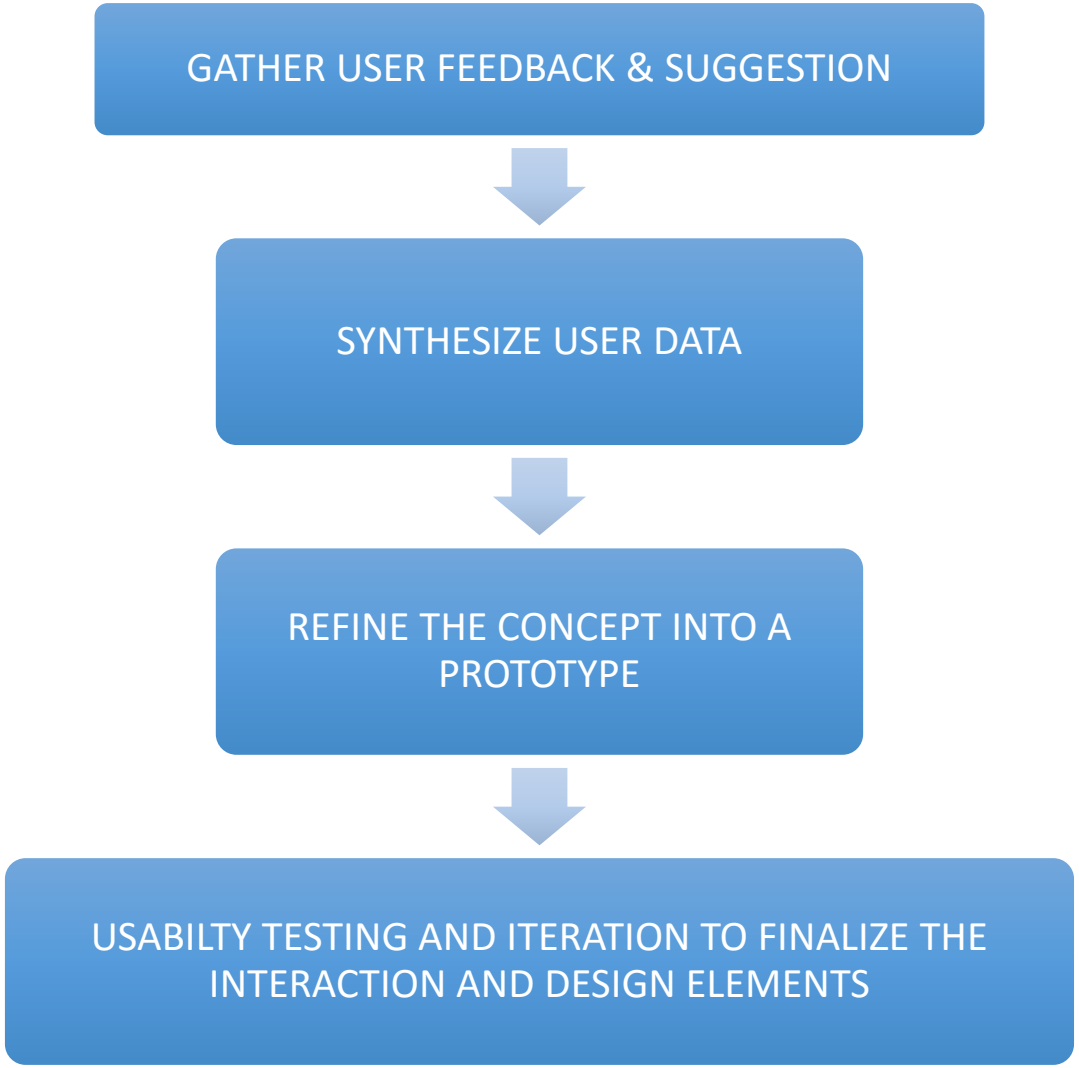
PRESENT THE 3 DESIGN ELEMENTS:

AUTO SUGGEST WIDGET

UPLOAD FUNCTION

TAG FEATURE

FOCUS GROUP


GATHER USER FEEDBACK & SUGGESTION
SYNTHESIZE USER DATA
REFINE THE CONCEPT INTO A PROTOTYPE
USABILTY TESTING AND ITERATION TO FINALIZE THE INTERACTION AND DESIGN ELEMENTS

TEST SCRIPT FOR PARTICIPANTS OF THE FOCUS GROUP

The 2 variations of the auto suggestion widget:
The first variation has the categories listed based on the most frequently used order. The second variation has the categories listed based on the alphabetical order.

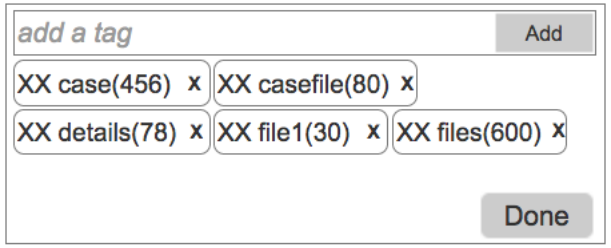
Variation 1

Variation 2

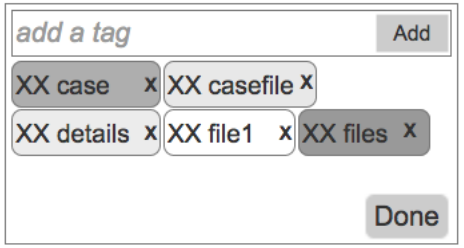
We will begin the focus group discussion by asking the participants about which variation they like more and why they prefer it and we will also request any alternatives that would work well for the scenario provided.

TEST SCRIPT FOR PARTICIPANTS OF THE FOCUS GROUP

The 2 variations of tagging option:
The first variation has numbers indicating the frequency of the tags. The second variation indicates frequency by color intensity



Variation 1



Variation 2

Through this discussion, we would like to know which one is more usable for the scenario provided and find further improvements

TEST SCRIPT FOR PARTICIPANTS OF THE FOCUS GROUP

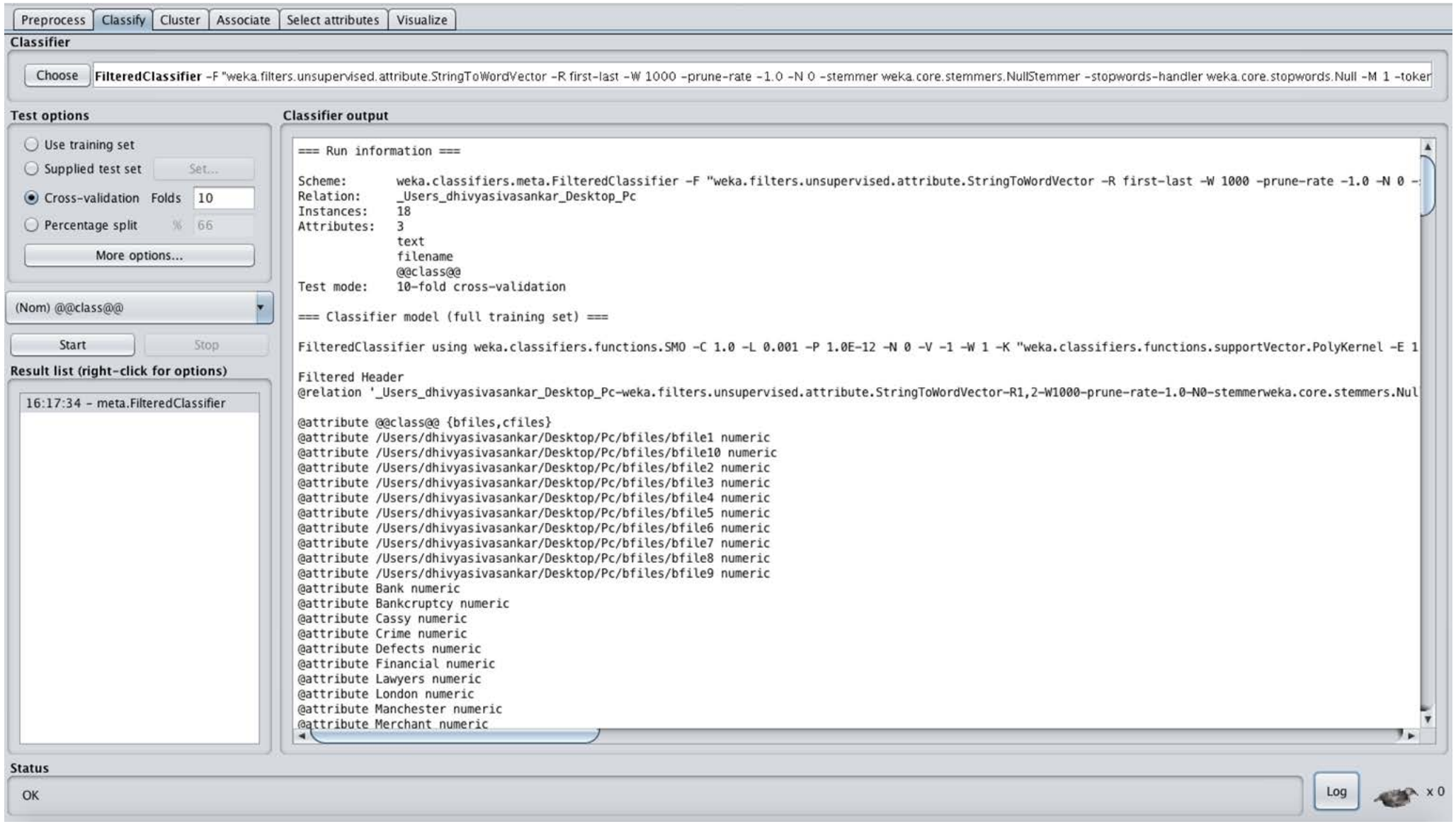
The 2 variations of the upload status display:
The first variation has pictorial representation of process steps on how the data will be stored, organized by the machine learning algorithm. The second variation just represent the percentage of the file that is being uploaded

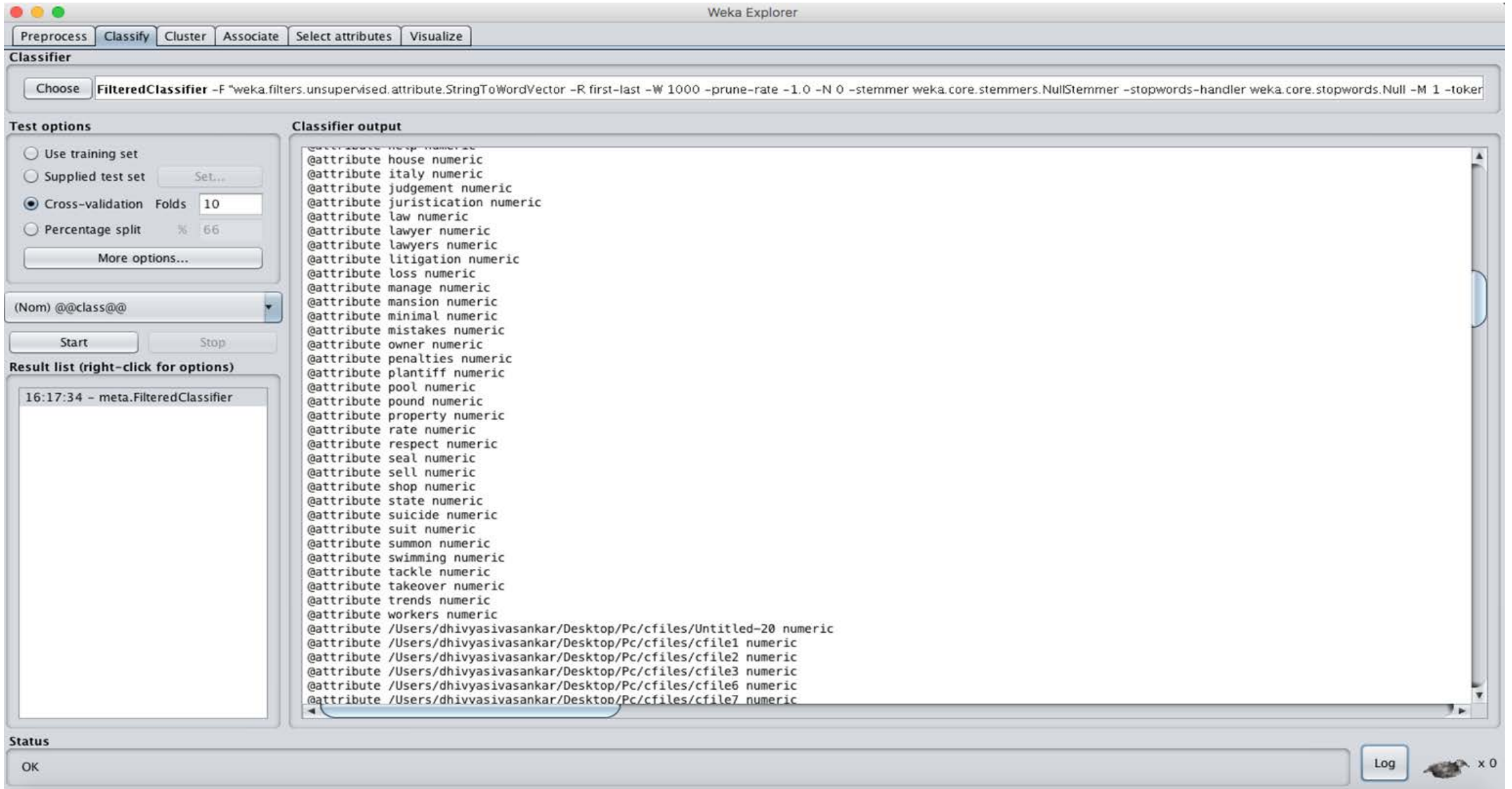
Variation 1

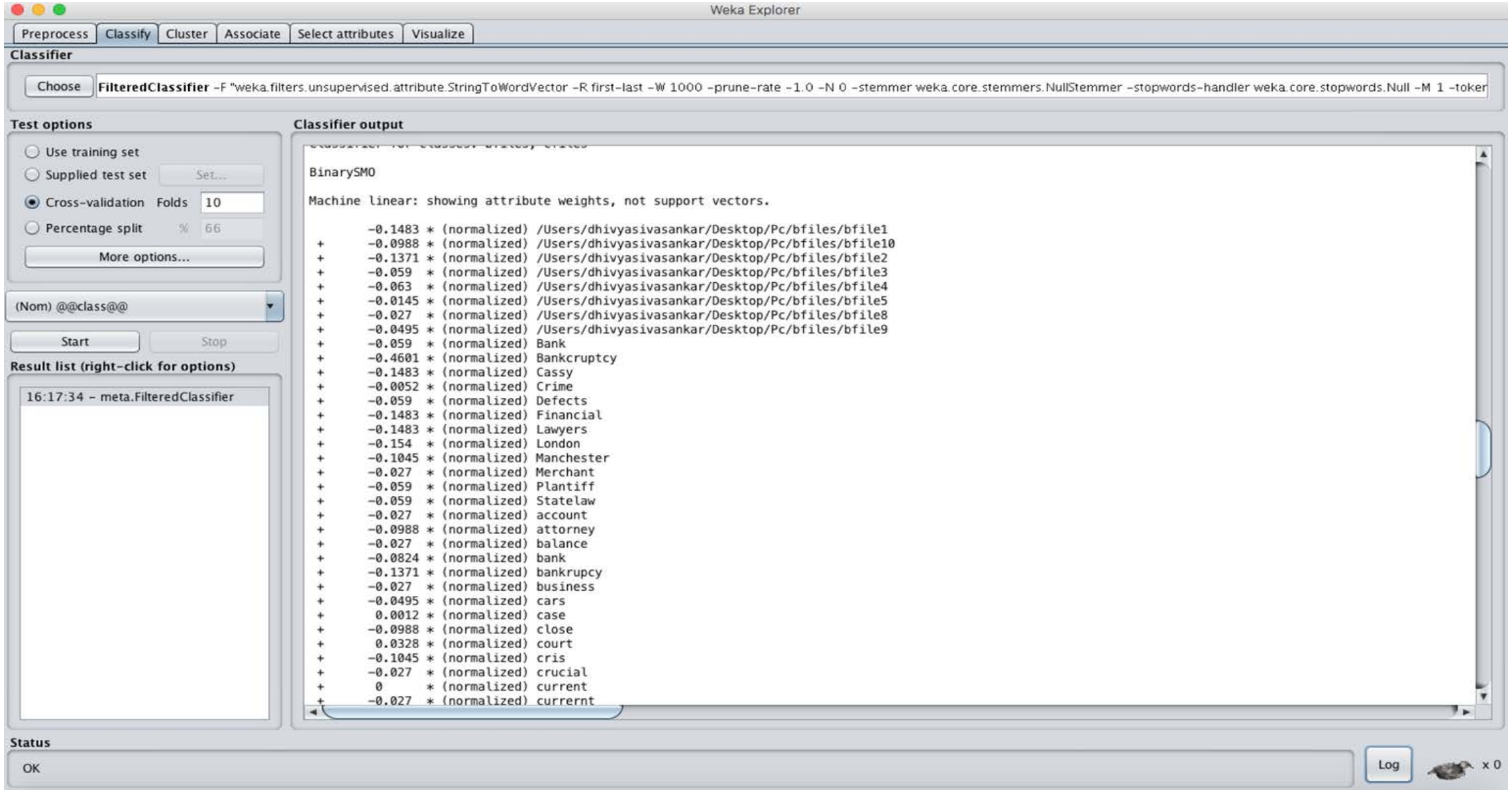
Variation 2

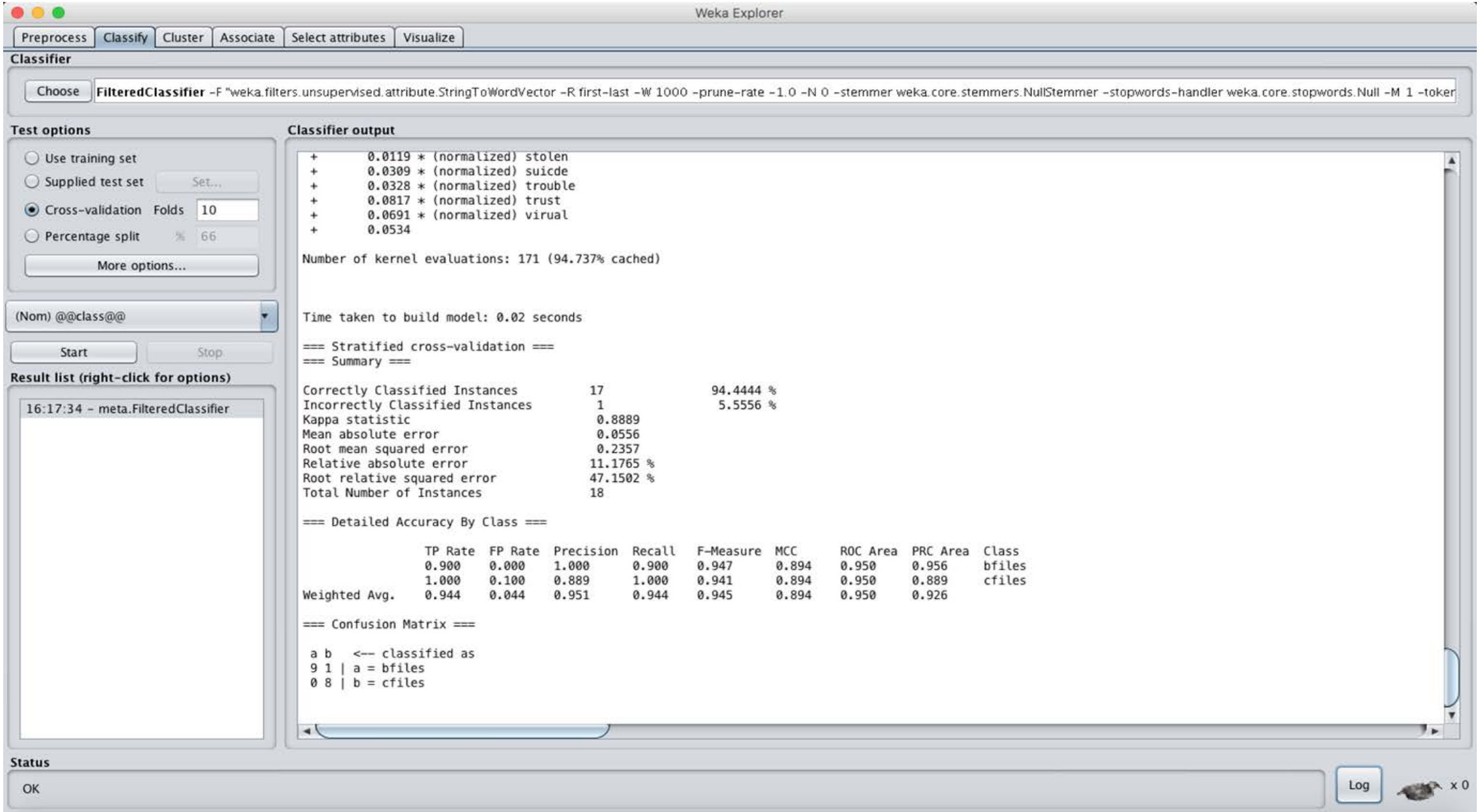
Through this discussion, we would like to know whether users find it useful to understand how their data is getting saved in the repository. For example, are these kind of options illustrating the backend process to them?

SCREEN SHOTS ANALYSIS RESULTS-WEKA TOOL




Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -toker
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) @@class@@
Start
Stop
Result list (right-click for options)
16:17:34 - meta.FilteredClassifier
Classifier output
@attribute house numeric
@attribute italy numeric
@attribute judgement numeric
@attribute juristication numeric
@attribute law numeric
@attribute lawyer numeric
@attribute lawyers numeric
@attribute litigation numeric
@attribute loss numeric
@attribute manage numeric
@attribute mansion numeric
@attribute minimal numeric
@attribute mistakes numeric
@attribute owner numeric
@attribute penalties numeric
@attribute plantiff numeric
@attribute pool numeric
@attribute pound numeric
@attribute property numeric
@attribute rate numeric
@attribute respect numeric
@attribute seal numeric
@attribute sell numeric
@attribute shop numeric
@attribute state numeric
@attribute suicide numeric
@attribute suit numeric
@attribute summon numeric
@attribute swimming numeric
@attribute tackle numeric
@attribute takeover numeric
@attribute trends numeric
@attribute workers numeric
@attribute /Users/dhivyasivasankar/Desktop/Pc/cfiles/Untitled-20 numeric
@attribute /Users/dhivyasivasankar/Desktop/Pc/cfiles/cfile1 numeric
@attribute /Users/dhivyasivasankar/Desktop/Pc/cfiles/cfile2 numeric
@attribute /Users/dhivyasivasankar/Desktop/Pc/cfiles/cfile3 numeric
@attribute /Users/dhivyasivasankar/Desktop/Pc/cfiles/cfile6 numeric
@attribute /Users/dhivyasivasankar/Desktop/Pc/cfiles/cfile7 numeric
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -toker
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) @@class@@
Start
Stop
Result list (right-click for options)
16:17:34 - meta.FilteredClassifier
Classifier output
BinarySMO
Machine linear: showing attribute weights, not support vectors.
−0.1483 * (normalized) /Users/dhivyasivasankar/Desktop/Pc/bfiles/bfile1
+ −0.0988 * (normalized) /Users/dhivyasivasankar/Desktop/Pc/bfiles/bfile10
+ −0.1371 * (normalized) /Users/dhivyasivasankar/Desktop/Pc/bfiles/bfile2
+ −0.059 * (normalized) /Users/dhivyasivasankar/Desktop/Pc/bfiles/bfile3
+ −0.063 * (normalized) /Users/dhivyasivasankar/Desktop/Pc/bfiles/bfile4
+ −0.0145 * (normalized) /Users/dhivyasivasankar/Desktop/Pc/bfiles/bfile5
+ −0.027 * (normalized) /Users/dhivyasivasankar/Desktop/Pc/bfiles/bfile8
+ −0.0495 * (normalized) /Users/dhivyasivasankar/Desktop/Pc/bfiles/bfile9
+ −0.059 * (normalized) Bank
+ −0.4601 * (normalized) Bankcruptcy
+ −0.1483 * (normalized) Cassy
+ −0.0052 * (normalized) Crime
+ −0.059 * (normalized) Defects
+ −0.1483 * (normalized) Financial
+ −0.1483 * (normalized) Lawyers
+ −0.154 * (normalized) London
+ −0.1045 * (normalized) Manchester
+ −0.027 * (normalized) Merchant
+ −0.059 * (normalized) Plantiff
+ −0.059 * (normalized) Statelaw
+ −0.027 * (normalized) account
+ −0.0988 * (normalized) attorney
+ −0.027 * (normalized) balance
+ −0.0824 * (normalized) bank
+ −0.1371 * (normalized) bankrupcy
+ −0.027 * (normalized) business
+ −0.0495 * (normalized) cars
+ 0.0012 * (normalized) case
+ −0.0988 * (normalized) close
+ 0.0328 * (normalized) court
+ −0.1045 * (normalized) cris
+ −0.027 * (normalized) crucial
+ 0 * (normalized) current
+ −0.027 * (normalized) currernt
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -toker
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) @@class@@
Start
Stop
Result list (right-click for options)
16:17:34 - meta.FilteredClassifier
Classifier output
+ 0.0119 * (normalized) stolen
+ 0.0309 * (normalized) suicde
+ 0.0328 * (normalized) trouble
+ 0.0817 * (normalized) trust
+ 0.0691 * (normalized) virual
+ 0.0534
Number of kernel evaluations: 171 (94.737% cached)
Time taken to build model: 0.02 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 17 94.4444 %
Incorrectly Classified Instances 1 5.5556 %
Kappa statistic 0.8889
Mean absolute error 0.0556
Root mean squared error 0.2357
Relative absolute error 11.1765 %
Root relative squared error 47.1502 %
Total Number of Instances 18
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.900 0.000 1.000 0.900 0.947 0.894 0.950 0.956 bfiles
1.000 0.100 0.889 1.000 0.941 0.894 0.950 0.889 cfiles
Weighted Avg. 0.944 0.044 0.951 0.944 0.945 0.894 0.950 0.926
=== Confusion Matrix ===
a b <-- classified as
9 1 | a = bfiles
0 8 | b = cfiles
Status
OK
Log
x 0

FUTURE WORK

- As of now, we are working with an ideal set of data that we created.
- Now we have to gather some real data sets.
- Also work on integrating the overall modules – database, logic, Natural Language Processing (NLP).
- Test with a focus group of lawyers in the field of bankruptcy.
- Obtain data sets in other areas of the law.

Any Questions?

Thank you for attending this session of HCI International 2017!

Please contact Professor Sara Anne Hook with questions, sahook@iupui.edu.